FR 010021

US

Europäisches Patentamt

European Patent Office

Office européen des brevets

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. **Patent application No.** **Demande de brevet n°**

01400504.5

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

DEN HAAG, DEN
THE HAGUE, 30/11/01
LA HAYE, LE

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 01400504.5
Demande n°:

Anmeldetag:
Date of filing: 27/02/01
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
Koninklijke Philips Electronics N.V.
5621 BA Eindhoven
NETHERLANDS

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Method for encoding MPEG-4 video data

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

"METHOD FOR ENCODING MPEG-4 VIDEO DATA"

FIELD OF THE INVENTION

The present invention relates to a method for encoding video data according to the MPEG-4 standard.

BACKGROUND OF THE INVENTION

The transmission of audio-visual data on lossy networks, such as the Internet or UMTS radio channels, requires the use of coding techniques that are both efficient in their use of bits and robust against transmission errors. The MPEG-4 standard, which has been designed in this context, exploits both the temporal and spatial redundancies found in natural and synthetic video sequences. To that purpose, for the three types of Video Object Planes (VOPs) present in the video stream (intra VOPs, or I VOPs ; predicted VOPs, or P VOPs ; and bidirectional VOPs, or B VOPs), specific coding techniques are used. These coding techniques, which reduce the bandwidth requirements by removing the redundancy in the video signal, become, when said signal is partitioned into packets in the transport layer, quite sensitive to bit errors and transport-layer packet losses (for example, a single bit error may make the decoding of information impossible, or, owing to a transport-layer packet loss, the predictively-coded motion information in the next packets may become undecodable). Moreover, due to the predictive nature of the encoder, an error which occurs in an I or P VOP tends to propagate to the following P and surrounding B VOPs.

Error resilience is then one of the numerous options offered by the MPEG-4 video standard in order to solve the previously indicated drawbacks. It provides a set of tools which allows to hierarchically sort out the encoded data according to their sensitivity. In order to take advantage of this feature, the transport layer must take into account the hierarchy information provided by the video layer.

The MPEG-4 video bitstreams are classically composed of a number of elements such as : Video Objects, Video Object Layers, Video Object Planes, Group of VOPs, Video Packets VPs, Video Data Partitions DPs, etc..., and MPEG-4 system manipulates entities such as : DecoderSpecificInfo, Access Units, SL packets. For this, to work properly, how to map the video elements into the system elements is of key importance. In particular, the mapping of video Data Partitions to system is described here. Video Data Partitions correspond to fragments of Video Packets, in a specific video bitstream syntax mode that enables them for error resilience purposes ; specifically, there are two video Data Partitions for each Video Packet. A drawback of the Data Partition syntax is however that it is not byte aligned, i.e. the boundary between the first and second Data Partitions of a Video Packet is not byte aligned : it does not start on a bit multiple of 8. This situation is sub-optimal for an efficient machine implementation and

may lead to problems when considering network transport, since network protocols transport bytes (i.e. slices of eight bits).

SUMMARY OF THE INVENTION

It is therefore the object of the invention to propose a method avoiding this drawback.

To this end, the invention relates to a method for encoding video data according to the MPEG-4 standard, in which, in order to map the video elements into the system elements and to avoid, in this case, any file formation interchange problem or any network problem, a specific alignment/fragmentation mechanism is chosen, according to which, when the video bit streams are encoded using the syntax mode corresponding to the fragmentation of the Video Object Planes (VOPs) contained in said video data into Video Packets (VPs), and Video Packets into Data Partitions (DPs), a video Data Partition is mapped into one or more SL packets, the first Video Data Partition start is always mapped to an SL packet start even if a large video Data Partition is splitted across several SL packets, and the last SL packet transporting the first Data Partition includes the separation marker (DC marker or Motion Marker depending on VOP type) and up to 7 subsequent bits of the second Data Partition in order to obtain byte alignment, the next SL packet starting on the next bit of the second Data Partition.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will now be described, by way of example, with reference to the accompanying drawings in which :

- Fig.1 gives the main processing layers of a multimedia terminal ;
- Fig.2 illustrates the aligment/fragmentation mechanism according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

The MPEG-4 standard provides standardized ways to represent audio-visual objects (called AVOs) of natural or synthetic origin; to compose them together to create compound AVOs that form audio-visual scenes, to multiplex and synchronize the data associated with AVOs, and to interact with the audio-visual scenes generated or reconstructed at the receiver's end. An audio-visual scene is generally composed of several AVOs, organized in a hierarchical fashion. The main processing stages of a multimedia audiovisual terminal allowing to render and display such an audiovisual scene is illustrated in Fig.1. This terminal is a multi-layer structure consisting of the three following layers : a TransMux layer 21, a FlexMux layer 22 and an Access Unit layer 23.

The TransMux layer 21 consists of a protection sublayer and a multiplexing sublayer (although it may not be possible to separately identify these sublayers in some

TransMux instances, the protection sublayer has a specific interest for providing error protection and error detection tools suitable for the given network or storage medium). This layer 21, not defined in the context of MPEG-4, is in fact an interface to the network or the storage medium and allows to offer transport services matching the requested Quality of Service (QoS). At its output, FlexMux streams are available, i.e. a sequence of FlexMux Packets (small data entities consisting of a header and a payload).

The FlexMux layer 22, completely specified by MPEG-4, consists of a flexible tool for interleaving data (one or more Elementary Streams into one FlexMux stream) and allows to identify the different channels for the data that have been multiplexed. At the output of said layer 22, SL-Packetized Streams are available, i.e. a sequence of SL-Packets that encapsulate one elementary stream (SL-Packet, or sync layer Packet = the smallest data entity managed by the next layer 23, or Sync layer, and comprising a configurable header and a payload itself consisting of a complete or partial access unit).

The layer 23, or Sync Layer, is provided to adapt elementary stream data for communication. The elementary streams are conveyed as SL-packetized streams, and this packetized representation additionnally provides timing and synchronization information, as well as fragmentation and random access information. This layer 23 is followed by the compression layer 24 which recovers data from its encoded format and allows to perform the necessary operations to decode the encoded signals and reconstruct the original information. This decoded information is then processed (composition, rendering) in view of its presentation (in order to be displayed) or of some user interactions.

According to the invention, the following solution is then proposed. When video bitstreams are encoded using the syntax mode corresponding to the fragmentation of VOPs into Video Packets, and Video Packets into Data Partitions, a Video Data Partition should be mapped into one or more SL packets. Specifically, it is allowed to split a large video Data Partition across several SL packets, but the first video data Partition start must always be mapped to an SL packet start.

Furthermore, since the second Data Partition is useless if the first one is lost but a decoder needs the marker to safely identify the end of the first Data Partition, the following alignment rule should be used : the last SL packet transporting the first Data Partition must include the separation marker (DC marker or Motion Marker, depending on VOP type) and up to 7 subsequent bits of the second Data Partition in order to obtain byte alignment. The next SL packet starts on the next bit of the second Data Partition. This alignment/fragmentation mechanism is illustrated in Fig.2.

CLAIM :

1. A method for encoding video data according to the MPEG-4 standard, in which, in order to map the video elements into the system elements and to avoid, in this case, any file formation interchange problem or any network problem, a specific alignment/fragmentation mechanism is chosen, according to which, when the video bit streams are encoded using the syntax mode corresponding to the fragmentation of the Video Object Planes (VOPs) contained in said video data into Video Packets (VPs), and Video Packets into Data Partitions (DPs), a video Data Partition is mapped into one or more Sync Layer packets (SL packets), the first Video Data Partition start is always mapped to an SL packet start even if a large video Data Partition is splitted across several SL packets, and the last SL packet transporting the first Data Partition includes the separation marker and up to 7 subsequent bits of the second Data Partition in order to obtain byte alignment, the next SL packet starting on the next bit of the second Data Partition.

Abstract

The invention relates to a method for encoding video data according to the MPEG-4 standard. In order to avoid any problem when mapping the video elements into the system elements, a specific alignment/fragmentation mechanism is chosen. According to this mechanism, when the video bit streams are encoded using the syntax mode corresponding to the fragmentation of the Video Object Planes (VOPs) contained in said video data into Video Packets (VPs), and Video Packets into Data Partitions (DPs), a video Data Partition is mapped into one or more SL packets (SLi), the first video Data Partition (DP1) start is always mapped to an SL packet start, and the last SL packet transporting the first Data Partition includes the separation marker and up to 7 subsequent bits of the second Data Partition (DP2) in order to obtain byte alignment, the next SL packet starting on the next bit of the second Data Partition.

Fig.2

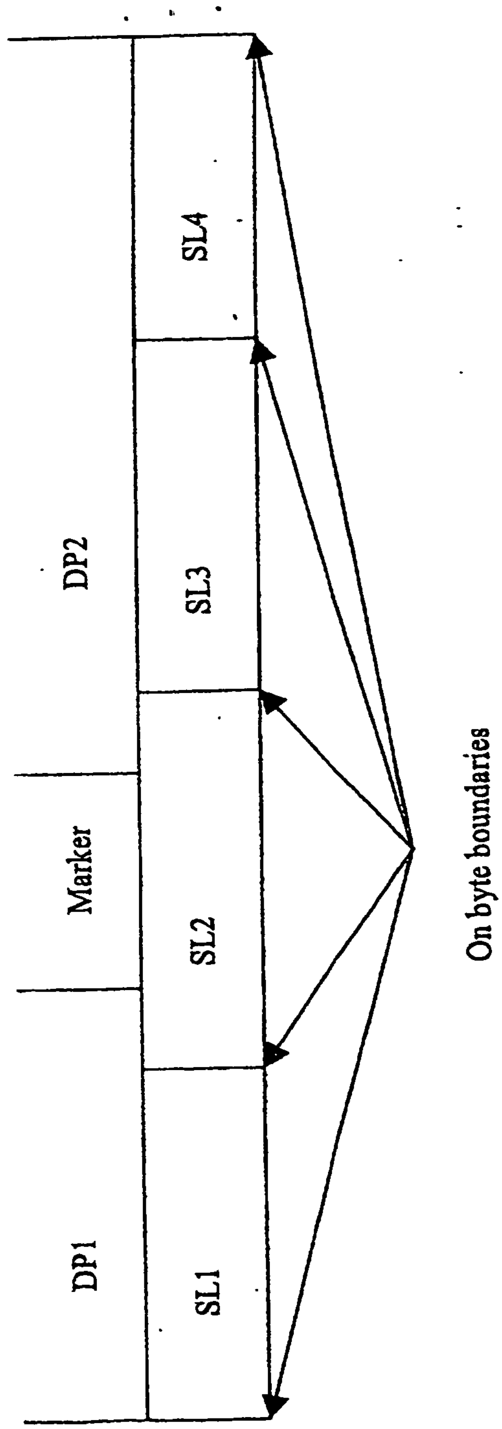
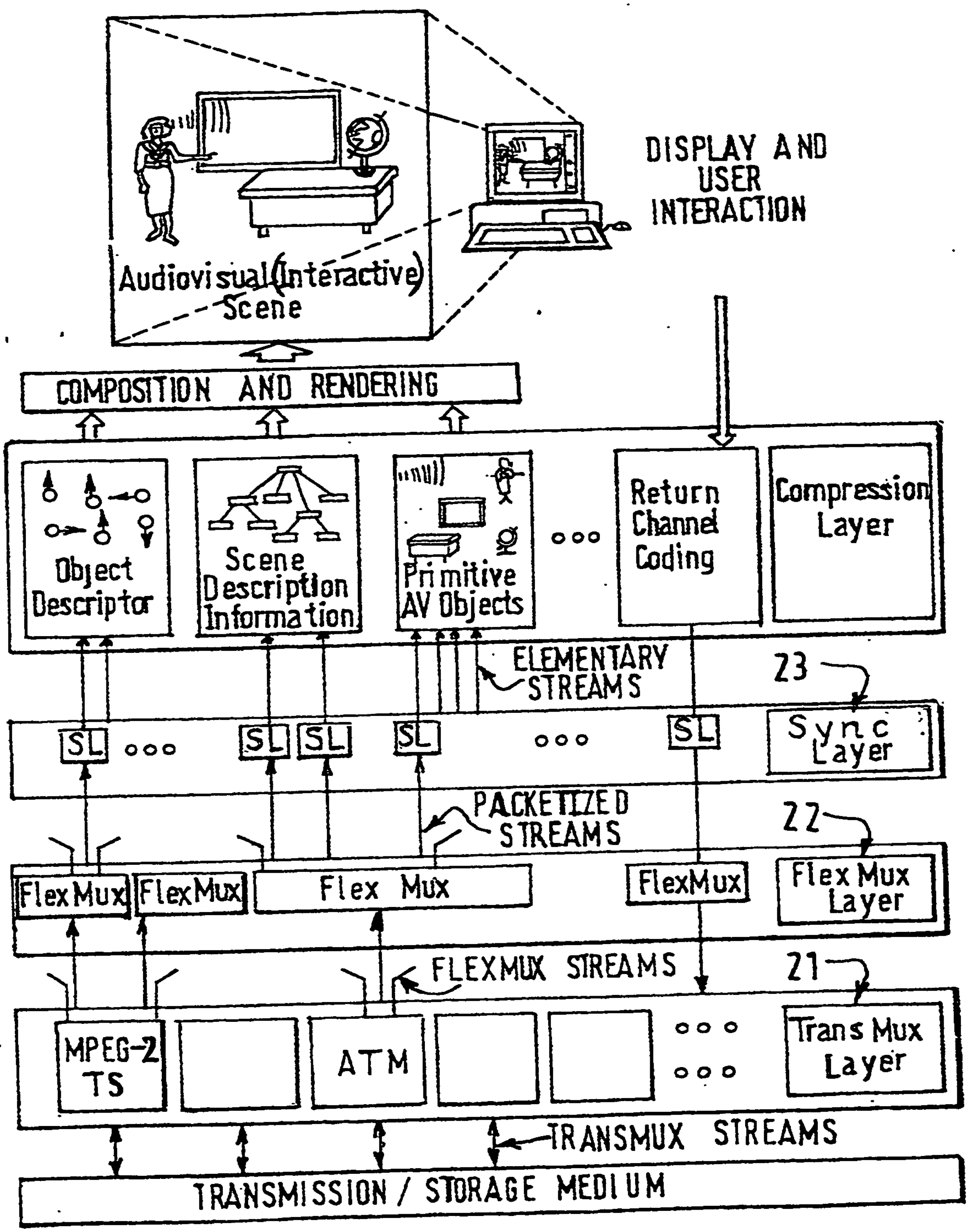


FIG. 2

1/2



DISPLAY AND USER INTERACTION
Audiovisual (Interactive) Scene
COMPOSITION AND RENDERING
Object Descriptor
Scene Description Information
Primitive AV Objects
Return Channel Coding
Compression Layer
ELEMENTARY STREAMS
23
SL
SL
SL
SL
SL
Sync Layer
PACKETIZED STREAMS
22
FlexMux
FlexMux
Flex Mux
FlexMux
Flex Mux Layer
FLEXMUX STREAMS
21
MPEG-2 TS
ATM
Trans Mux Layer
TRANSMUX STREAMS
TRANSMISSION / STORAGE MEDIUM

FIG.1

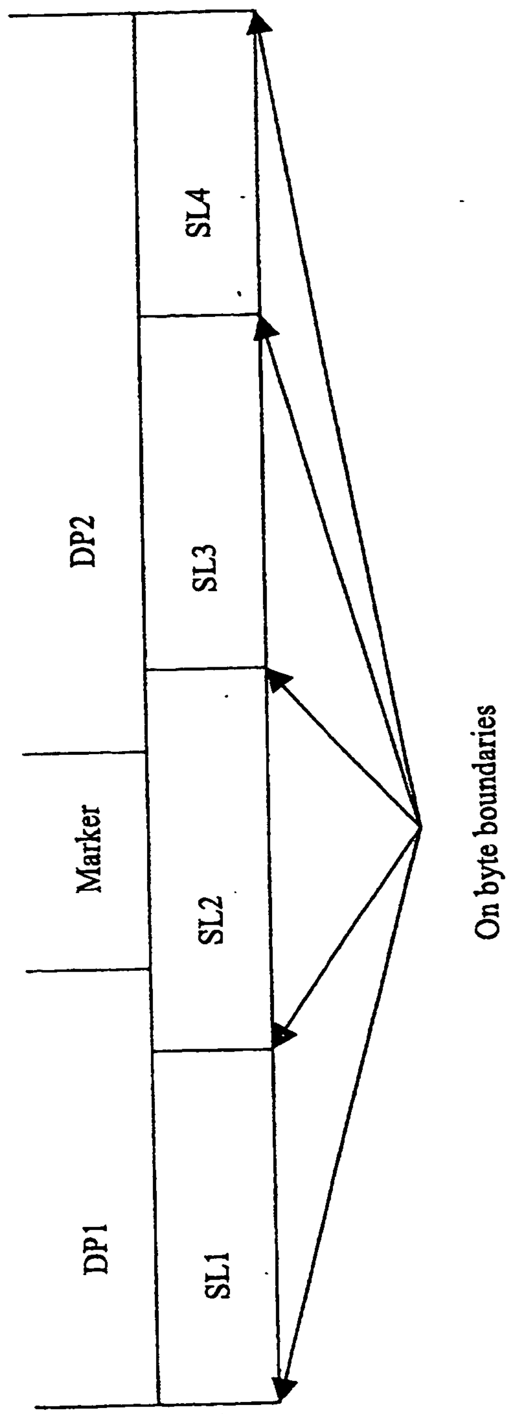


FIG. 2